0.80535

Objective

The goal of this project was to develop a regression-based classification model that predicts the likelihood of colorectal cancer using physical examination indicators. The dataset includes biochemical, hematological, and fecal test results. Our primary objective was to build an interpretable model with a strong F1 score to help identify high-risk individuals early.

Procedure

Dataset and Preprocessing

The original dataset contained 45 features (X1–X45) and a binary outcome variable y. After inspection, we found X5 to be a suspicious duplicate of Glucose with abnormal concentration around exactly 6.0 and rare outliers up to 60.0, so it was dropped.

We imputed missing values using the median strategy and applied winsorization to handle outliers. Skewed features (|skewness| > 1) were log-transformed to improve symmetry:

```
                  Applying log1p to 22 skewed features:
 ['X25', 'X8', 'X9', 'X21', 'X20', 'X18', 'X17', 'X38', 'X19', 'X4', 'X23',
'X11', 'X30', 'X45', 'X10', 'X29', 'X12', 'X26', 'X31', 'X28', 'X44', 'X34']
```

Feature Engineering and Interaction Terms

After preprocessing, we constructed 12 medically meaningful interaction terms based on medical understanding, including Creatinine × Urea (kidney), ALP × GGT (liver), Hemoglobin × Fecal Occult Blood (bleeding risk), and others. This helped capture relationships between features that may signal colorectal cancer.

Honestly, due to my limited knowledge of medicine, I kindly asked ChatGPT to help me with those interaction terms. I have to admit that they did a great job!

```
interaction_pairs = [
('X6', 'X7'),  # Creatinine * Urea — kidney function
('X12', 'X13'), # Direct Bilirubin * Indirect Bilirubin — liver function
('X14', 'X15'), # Total Protein * Albumin — protein metabolism
('X16', 'X17'), # A/G Ratio * ALT — liver stress marker
('X19', 'X21'), # ALP * GGT — cholestasis
('X26', 'X27'), # WBC * Lymphocytes — immune response
('X32', 'X33'), # RBC * Hemoglobin — anemia evaluation
('X36', 'X37'), # MCH * MCHC — red blood cell indices
('X40', 'X41'), # MPV * PCT — platelet volume and percentage
('X19', 'X45'), # ALP * Fecal Transferrin
('X33', 'X44'), # Hemoglobin * Fecal Occult Blood
('X8', 'X44')  # CEA * Fecal Occult Blood
]
```

Modeling Approach

We explored multiple models, including:

1. L1-penalized Logistic Regression
   Interpretable and used for baseline comparison and feature selection. However, its F1 score was lower than that of our final model.
2. Generalized Additive Model (GAM)
   Our best model. It captured non-linear effects via splines (s(i)) and treated binary variables X44 and X45 as categorical factors (f(i)).

We dynamically built GAM terms using column names to avoid index mismatch errors, especially after transformations and feature additions.

Hyperparameter Tuning and Threshold Optimization

We used a grid search over n_splines ∈ {5, 10, 15} and lam ∈ {0.1, 1, 10} to select the best GAM configuration based on validation F1 score.

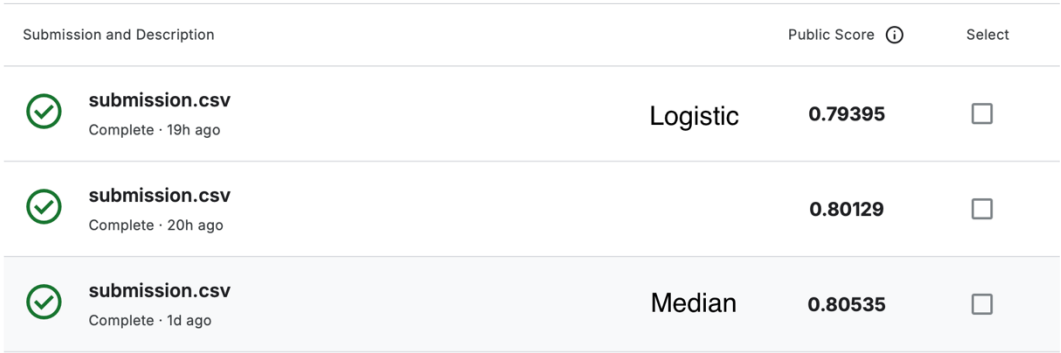
After fitting the model, we optimized the classification threshold over a range from 0.2 to 0.8. This allowed us to balance false positives and false negatives more effectively, especially given the medical context.

Trials, Errors and Insights

Initially, I planned to generate interaction terms automatically by identifying highly correlated feature pairs, a common technique for capturing nonlinear and synergistic relationships. I computed Pearson correlation coefficients for all feature pairs and selected the top three to create interaction terms. However, this approach led to a decrease in the F1 score, prompting me to switch to the current method—constructing interaction terms based on medical expertise.

I also experimented with using PCA for dimensionality reduction. Although it didn't improve the model's predictive performance as hoped, it did help streamline the modeling process by significantly reducing complexity and saving time during hyperparameter tuning.

To handle missing values, I initially experimented with using logistic regression to predict the missing entries in columns X44 and X45. I excluded X44, X45, and X5 (the weird feature) and used the remaining variables as predictors. Surprisingly, this approach did not yield better performance than the simpler median imputation strategy. This outcome highlights that, sometimes, straightforward methods can be just as effective—if not more so—than complex ones.



Comparison with Other Models

While logistic regression (L1) offered some interpretability and baseline performance, its predictive power lagged behind GAM. GAM provided a more flexible fit and better captured complex relationships in the data, improving F1 score from ~0.78 to ~0.80 after tuning and threshold adjustment.

Identifying Influential and Redundant Features

To better understand which variables most strongly affect colorectal cancer prediction, I used both statistical and interpretive methods based on the final fitted Generalized Additive Model (GAM). Each term in the model was evaluated by three key criteria:

1. Partial dependence-based influence score — computed as the average absolute gradient of the partial dependence function, which quantifies how sensitively the predicted probability changes with a feature.

```
Top 10 most influential features:
inter_X33_X44 (term 35): influence score = 0.0496
X43 (term 25): influence score = 0.0408
```

2. Statistical significance — retrieved from `model.summary()`, particularly the p-value column (P > x), which tests whether each term contributes significantly to the model.
3. Effective degrees of freedom (EDoF) — indicating the complexity used by the spline smoother for each feature.

Feature Function	Lambda	Rank	EDoF	P > x
s(0)	[10]	15	7.1	2.07e-03
s(1)	[10]	15	6.1	2.19e-02
s(2)	[10]	15	5.7	3.49e-03
s(3)	[10]	15	5.8	2.36e-01
s(4)	[10]	15	5.4	0.00e+00
s(5)	[10]	15	5.5	8.47e-05
s(6)	[10]	15	6.1	2.10e-08
s(7)	[10]	15	5.8	1.78e-03
s(8)	[10]	15	5.2	5.23e-03
s(9)	[10]	15	5.7	5.19e-02
s(10)	[10]	15	5.7	4.05e-11
s(11)	[10]	15	5.9	1.11e-01
s(12)	[10]	15	5.8	2.01e-12
s(13)	[10]	15	5.4	3.15e-02
s(14)	[10]	15	2.7	0.00e+00
s(15)	[10]	15	5.4	7.05e-07
s(16)	[10]	15	5.5	1.24e-03

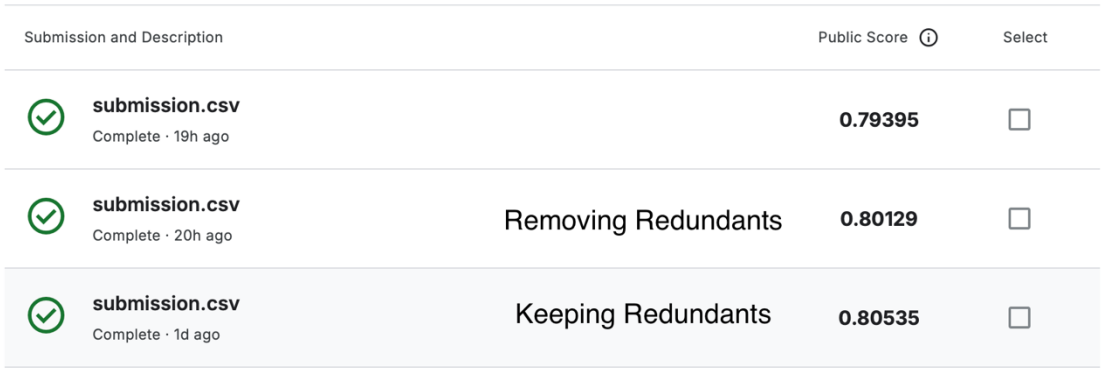
The most influential feature identified was the interaction term inter_X33_X44, with an influence score of 0.0496, a p-value of 4.48e-11, and an EDoF of 3.1. This indicates both strong nonlinear influence and high statistical significance in the model.

To identify redundant or low-impact features, I filtered out all GAM terms with p-values greater than 0.2. This led to the removal of 20+ features including X3, X5, X13, and X16 among others. These features had consistently low influence scores, low complexity (EDoF), and failed the significance test.

```
Dropping the following features due to high p-value (> 0.2):
   Feature Function    P > x
2            s(2)  0.345211
4            s(4)  0.310520
11          s(11)  0.692725
12          s(12)  0.892006
13          s(13)  0.260091
14          s(14)  0.391519
15          s(15)  0.906629
17          s(17)  0.220362
21          s(21)  0.620874
24          s(24)  0.396222
25          s(25)  0.441381
26          s(26)  0.369548
27          s(27)  0.588196
30          s(30)  0.340374
39          s(39)  0.609262
40          s(40)  0.569902
42          s(44)  0.336099
45          s(47)  0.694087
50          s(52)  0.438464
```

Impact of Removing Redundant Features

After removing statistically insignificant features (p > 0.2), I retrained the GAM. The F1 score on the test set dropped slightly from 0.80535 to 0.80129. While this suggests that the removed features had minor predictive value, the change was marginal and not practically significant.



However, the benefit was substantial in terms of computational efficiency. The original model, which included all 50+ terms, took over 5 minutes to train and predict. After filtering, the training and prediction time dropped to under 1 minute — a major gain in speed, especially useful during iterative tuning.

This trade-off — a minor loss in F1 for a large gain in runtime — supports the idea that simplifying a GAM via significance-based pruning is viable, especially when performance

saturation has been reached.

Consequences of False Positives and Negatives

In a medical setting, false negatives are more dangerous—they represent missed cancer diagnoses. A false positive, while inconvenient, prompts further testing and can be corrected.
To address this, we selected a threshold that maximized F1 score in our course project, which balances both types of error. However, in real deployment, I'd recommend that we prioritize recall more strongly because we don't want any diagnosis to be missed.

Overfitting and Generalization

We monitored performance across training and validation sets to detect overfitting. Early models without spline terms showed signs of overfitting (a ~0.2 f1 gap between validation set and test set).
Then we improved generalization by introducing the splines. A grid of 9 models with the number of segments in [5, 10, 15] and smooth penalty in [0.1, 1, 10] was introduced. Those models compete with each other and the best wins. To be specific, the winner is:

```
Best n_splines: 15, Best lam: 10
```

There are also some insights from the mistakes I made:
Dummy variables must be treated carefully
I initially made the mistake of passing binary features like X44 and X45 through spline terms. Replacing them with categorical f(i) terms resolved model errors and improved performance.
Column alignment is critical
After adding interaction terms, keeping feature order consistent across train/val/test sets avoided hard-to-debug index mismatch errors.

Future data collection

Avoid Missing Values
If we want a better prediction, we should have a complete dataset at first place. In our current training data, some key diagnostic indicators, such as fecal occult blood (X44) and fecal transferrin (X45), have millions of missing values. Many of the interaction terms contain those features and some of them are among the most influential predictors in our model. This introduced not only uncertainty but also processing complexity. Thus, I would want those markers fully recorded.
Include Personal Information
Also, we should expand the dataset to include personal information like age, gender, lifestyle habits and family history. While the current model uses only numerical lab results, colorectal cancer risk is known to be influenced by several demographic and behavioral factors. Including these variables would offer a more holistic view of patient risk and likely boost model performance by capturing risk patterns not visible in lab data alone.

Drop Suspicious Features

Finally, I recommend reviewing the data integrity and measurement protocols for duplicated or suspicious features—specifically X5, which was intended as a second glucose measurement. This variable contained mostly identical values with occasional extreme outliers, raising concerns about data entry or sensor calibration. To avoid introducing noise or misleading the model, future datasets should enforce stricter validation on repeated or critical lab indicators, or clearly flag when a secondary reading is unavailable or invalid.

Our Diagnosis

Now here comes two patients, Mrs. Six and Mr. Seven. According to our model, both received a “Negative”. However, as professionals, we don’t let even the slightest possibility slide, so we take a closer look at the actually prediction scores of the GAM model:

```
Final probabilities patient 6 and 7: 0.12055340998469796 0.030964503611746932
```

Now let’s compare these to our decision threshold:

```
Best threshold: 0.38000000000000006
```

Both of their GAM scores are way below the threshold, so they can leave with a relief. However, out of an abundance of caution, we’d advise them that if they have relevant symptoms, a family history of the condition, or simply wish to be proactive, it would still be wise to consider additional testing.